



000100100
0001100110
01001010000
11011011000

4^e édition 2001

JavaTM 2

Le guide du développeur

Inclut J2SDK 1.3
pour Windows et Linux



Formation directe au premier langage du troisième millénaire !

Aucune expérience requise - Résultats assurés

Nouveau ! JDBC et XML inclus

Pierre-Yves Saumont et Antoine Mirecourt

www.oemweb.com



OSMAN EYROLLES MULTIMEDIA



2-005-594-1



Sommaire

Java 2

Le guide du développeur

Pierre-Yves Saumont
et Antoine Mirecourt



Sommaire



Introduction	xxv
Pourquoi Java ?	xxv
Qu'est-ce que Java ?	xxx
Java est un langage orienté objets	xxx
Java est extensible à l'infini	xxxiii
Java est un langage à haute sécurité	xxxiv
Java est un langage simple à apprendre	xxxv
Les compilateurs natifs	xxxvii
Les compilateurs de bytecode	xxxviii
Les compilateurs JIT	xxxix
La technologie HotSpot	xl
Le premier langage du troisième millénaire ?	xli
Voir Java à l'œuvre	xlii
Support technique	xlii
Aller plus loin	xlvi
Le Développeur Java 2, Mise en œuvre et solutions - Les applets	xlvi
Servlets et Java Server Pages, le guide du développeur	xlvi

Chapitre 1 :**Installation de Java**

	1
Configuration nécessaire	3
Ordinateur	3
Système	3
Mémoire	4
Disque	4
Installation du J2SDK sous Linux	4
Installation de la version auto-extractible	4
Installation de la version rpm	5
Installation du plug-in Java 2	6
Installation de la documentation sous Linux	7
Configuration du J2SDK sous Linux	8
Installation du J2SDK sous Windows	8
Configuration de l'environnement sous Windows	11
Le chemin d'accès aux programmes exécutables sous Windows 9x	11
Le chemin d'accès aux programmes exécutables sous Windows NT et Windows 2000	13
Le chemin d'accès aux classes Java	14
Ce que contient le J2SDK	19
Le "Java 2 Runtime Environment"	27
Installation du J2RE Windows	29
Installation de la documentation sous Windows	31
Un navigateur HTML compatible Java	33
HotJava	33
Netscape Navigator	34
Netscape Navigator 6	35
Internet Explorer	35
Un éditeur pour la programmation	36
Ce que doit offrir un éditeur	37
UltraEdit	38
Quelques petits fichiers batch qui vous simplifieront la vie	38
Un environnement de développement	40
Avantages	40

Inconvénients	42
Quel environnement ?	42
Forte for Java	43
Où trouver les informations ?	44
Le site Web de Sun	44
Le JDC	44
Les autres sites Web dédiés à Java	45
Le site du "Développeur Java 2"	46
Les Newsgroups	46
Chapitre 2 :	
Votre premier programme	47
Première version : un programme minimal	48
Analyse du premier programme	52
Première applet	61
Analyse de la première applet	63
Une façon plus simple d'exécuter les applets	68
Résumé	69
Exercice	70
Chapitre 3 :	
Les objets Java	71
Tout est objet	74
Les classes	74
Pertinence du modèle	76
Les instances de classes	79
Les membres de classes	79
Les membres statiques	81
Les constructeurs	83
Création d'un objet	85
La destruction des objets : le garbage collector	89
Comment retenir les objets : les handles	90

Création des handles	92
Modifier l'affectation d'un handle	92
Résumé	94
Exercice	94

Chapitre 4 :

Les primitives et les handles

Les primitives	99
Utiliser les primitives	100
Valeurs par défaut des primitives	104
Différences entre les objets et les primitives	106
Les valeurs littérales	110
Le casting des primitives	112
Le casting explicite	113
Casting d'une valeur entière en valeur flottante	114
Casting d'une valeur flottante en valeur entière	115
Formation des identificateurs	115
Portée des identificateurs	117
Portée des identificateurs de méthodes	122
Les objets n'ont pas de portée	123
Les chaînes de caractères	123
Constantes	127
Utilisation de final avec des objets	127
Accessibilité	128
Retour sur les variables statiques	128
Masquage des identificateurs de type static	134
Java et les pointeurs	137
Exercices	138
Première question (page 115)	138
Deuxième question (page 136)	139
Résumé	140

Chapitre 5 :**Créez vos propres classes**

Tout est objet (bis)	142
L'héritage	143
Les constructeurs	145
Référence à la classe parente	148
La redéfinition des méthodes	148
Surcharger les méthodes	150
Signature d'une méthode	151
Optimisation	153
Surcharger les constructeurs	156
Les constructeurs par défaut	159
Les initialiseurs	165
Les initialiseurs de variables d'instances	166
Les initialiseurs d'instances	167
Ordre d'initialisation	168
Méthodes : les valeurs de retour	171
Surcharger une méthode avec une méthode de type différent	174
Distinction des méthodes surchargées par le type uniquement	175
Le retour	177
Résumé	178
Exercice	179

Chapitre 6 :**Les opérateurs**

Java et les autres langages	184
L'opérateur d'affectation	185
Les opérateurs arithmétiques à deux opérandes	187
Les opérateurs à un opérande	191
Les opérateurs relationnels	194
Les opérateurs logiques	201
Les opérateurs d'arithmétique binaire	204

L'opérateur à trois opérandes ?:	216
Les opérateurs de casting	217
L'opérateur new	219
L'opérateur instanceof	219
Priorité des opérateurs	220
Résumé	223

Chapitre 7 :

Les structures de contrôle

La séquence	226
Le branchement par appel de méthode	226
L'instruction de branchement return	228
L'instruction conditionnelle if	229
L'instruction conditionnelle else	231
Les instructions conditionnelles et les opérateurs ++ et --	232
Les instructions conditionnelles imbriquées	233
La boucle for	236
L'initialisation	237
Le test	238
L'incrémementation	239
Le bloc d'instructions	240
Modification des indices à l'intérieur de la boucle	240
Imbrication des boucles for	241
Type des indices	242
Portée des indices	243
Sortie d'une boucle par return	246
Branchement au moyen des instructions break et continue	246
Utilisation de break et continue avec des étiquettes	248
L'instruction while	249
L'instruction switch	254
L'instruction synchronized	260
Résumé	261

Chapitre 8 :**L'accessibilité**

263

Les packages (Où)	267
Chemins d'accès et packages	270
L'instruction package	270
Placement automatique des fichiers .class dans les répertoires correspondant aux packages	272
Exécution du programme	273
Chemin d'accès par défaut pour les packages	274
L'instruction import	275
Packages accessibles par défaut	278
Les fichiers .jar	279
Création de vos propres fichiers .jar ou .zip	280
Comment nommer vos packages ?	281
Ce qui peut être fait (Quoi)	282
static	283
Les variables static	283
Les méthodes static	285
Initialisation des variables static	286
Les initialiseurs statiques	287
final	288
Les variables final	289
Les variables final non initialisées	290
Les méthodes final	291
Les classes final	292
synchronized	292
native	293
transient	294
volatile	294
abstract	294
Les interfaces	296
Qui peut le faire (Qui)	297
public	297
protected	298

<i>package</i>	298
<i>private</i>	298
Autorisations d'accès aux constructeurs	299
Résumé	304

Chapitre 9 :

Le polymorphisme

Le sur-casting des objets	306
Retour sur l'initialisation	308
Le sur-casting	310
Le sur-casting explicite	313
Le sur-casting implicite	315
Le sous-casting	315
Le "late binding"	317
Les interfaces	322
Utiliser les interfaces pour gérer des constantes	323
Un embryon d'héritage multiple	325
Le polymorphisme et les interfaces	326
Pas de vrai héritage multiple	329
Quand utiliser les interfaces ?	332
Héritage et composition	335
Résumé	338

Chapitre 10 :

Les tableaux et les collections

Les tableaux	339
Déclaration	340
Initialisation	341
Initialisation automatique	342
Les tableaux littéraux	343
Les tableaux de primitives sont des objets	344
Le sur-casting implicite des tableaux	345

Les tableaux d'objets sont des tableaux de handles	348
La taille des tableaux	351
Les tableaux multidimensionnels	353
Les tableaux et le passage d'arguments	355
Copie de tableaux	357
Les vecteurs	360
Le type Stack	364
Le type BitSet	365
Les tables (Map)	366
Les tables ordonnées (SortedMap)	367
Le type Hashtable	368
Les collections	369
Les listes (List)	370
Les ensembles (Set)	372
Les ensembles ordonnés (SortedSet)	372
Les itérateurs	373
Les itérateurs de listes	375
Les comparateurs	375
Les méthodes de la classe Collections	382
Exemple : utilisation de la méthode sort()	385
Résumé	387
Chapitre 11 :	
Les objets meurent aussi	389
<hr/>	
Certains objets deviennent inaccessibles	389
Que deviennent les objets inaccessibles ?	395
Le garbage collector	395
Principe du garbage collector	396
Optimiser le travail du garbage collector	398
Les finaliseurs	399
Contrôler le travail du garbage collector	403
Références et accessibilité	406
Les références faibles	407
SoftReference	410

WeakReference	410
PhantomReference	411
Les queues	411
Exemple d'utilisation de références faibles	412
Autres formes de contrôle du garbage collector	415
La finalisation et l'héritage	421
La finalisation et le traitement d'erreur	422
Contrôler le garbage collector à l'aide des options de l'interpréteur	422
Résumé	423

Chapitre 12 :

Les classes internes

Les classes imbriquées	425
Les classes membres	432
Instances externes anonymes	439
Classes membres et héritage	440
Remarque concernant les classes membres	443
Les classes locales	446
Les handles des instances de classes locales	448
Les classes anonymes	452
Comment sont nommées les classes anonymes	454
Résumé	455

Chapitre 13 :

Les exceptions

Stratégies de traitement des erreurs	457
Signaler et stopper	458
Corriger et réessayer	459
Signaler et réessayer	459
La stratégie de Java	459
Les deux types d'erreurs de Java	460
Les exceptions	461

Attraper les exceptions	463
Dans quelle direction sont lancées les exceptions ?	464
Manipuler les exceptions	465
Modification de l'origine d'une exception	467
Créer ses propres exceptions	474
La clause finally	478
Organisation des handlers d'exceptions	481
Pour un véritable traitement des erreurs	483
D'autres objets jetables	484
Les exceptions dans les constructeurs	485
Exceptions et héritage	485
Résumé	485
Chapitre 14 :	
Les entrées/sorties	487
Principe des entrées/sorties	488
Les streams de données binaires	489
Les streams d'entrée	489
Streams de communication	489
Streams de traitement	489
Les streams de sortie	490
Streams de communication	490
Streams de traitement	490
Les streams de caractères	492
Les streams d'entrée	492
Streams de communication	492
Streams de traitement	493
Les streams de sortie	493
Streams de communication	494
Streams de traitement	494
Les streams de communication	494
Lecture et écriture d'un fichier	496
Les streams de traitement	497
Exemple de traitement : utilisation d'un tampon	502

Exemple de traitement : conversion des fins de lignes	503
Compression de données	504
Décompression de données	509
La sérialisation	509
Les fichiers à accès direct	512
Résumé	516

Chapitre 15 :

Le passage des paramètres

Passage des paramètres par valeur	517
Passage des paramètres par référence	518
Passer les objets par valeur	520
Le clonage des objets	521
Clonage de surface et clonage en profondeur	524
Clonage en profondeur d'un objet de type inconnu	529
Clonabilité et héritage	529
Interdire le clonage d'une classe dérivée	532
Une alternative au clonage : la sérialisation	533
Une autre alternative au clonage	536
Résumé	538

Chapitre 16 :

Exécuter plusieurs processus simultanément

Qu'est-ce qu'un processus ?	540
Avertissement	540
Comment fonctionnent les threads	541
Principes de synchronisation	545
Les opérations atomiques	548
Granularité de la synchronisation	550
Atomicité apparente	551
Créer explicitement un thread	551
Exécuter plusieurs threads simultanément	553

Caractéristiques des threads	558
Contrôler les threads	559
Utilisation des groupes pour référencer les threads	561
Gérer la répartition du temps entre les threads	565
La priorité	568
La synchronisation	569
Problèmes de synchronisation	575
Mise en œuvre d'un démon	584
Communication entre les threads : wait et notifyAll	586
Résumé	590
Exercice	590

Chapitre 17 :

RTTI et réflexion	593
-------------------	-----

Le RTTI ou comment Java vérifie les sous-castings explicites	593
Connaître la classe d'un objet	595
Instancier une classe à l'aide de son objet Class	599
Connaître la classe exacte d'un objet	599
Utiliser la réflexion pour connaître le contenu d'une classe	600
Utilité de la réflexion	600
Utiliser la réflexion pour manipuler une classe interne	606
Utiliser la réflexion pour créer des instances	607
Conclusion : quand utiliser la réflexion ?	607

Chapitre 18 :

Fenêtres et boutons	609
---------------------	-----

Les composants lourds et les composants légers	610
Les composants lourds	610
Les composants légers	611
Le look & feel	611
Les fenêtres	613
Hiérarchie des fenêtres Java	613

Structure d'une fenêtre	614
Les layout managers	617
Créer une application fenêtrée	618
Quel événement intercepter ?	619
Intercepter les événements de fenêtre	620
Les événements et les listeners	620
Utilisation des composants	629
Utilisation des layout managers	631
Philosophie des layout managers	631
Bien utiliser les layout managers	636
Utilisation du FlowLayout	640
Utilisation d'un layout à plusieurs niveaux	646
Utilisation du GridBagLayout	649
Rendre l'interface réellement indépendante du système	655
Affichage sans layout manager	657
Créer votre propre layout manager	659
Les éléments de l'interface utilisateur	661
Les boutons	662
Les événements de souris	668
Les menus	674
Les fenêtres internes	678
Le plaf (Pluggable Look And Feel)	683
Exemples de composants	690
Résumé	701

Chapitre 19 :

Le graphisme

703

Les primitives graphiques	703
Les objets graphiques	704
Un composant pour les graphismes	704
Les différentes primitives	709
L'affichage du texte	722
Les polices de caractères	723
Les images	730

Obtenir une image	730
Surveiller le chargement d'une image	733
Conclusion	739

Chapitre 20 :

Applets et réseaux	741
--------------------	-----

Les applets	741
Création d'une applet	742
Problèmes de compatibilité entre les versions	743
Avertissement	744
Deuxième avertissement	745
Fonctionnement d'une applet	745
Passer des paramètres à une applet	746
Agir sur le navigateur	750
Afficher un nouveau document	750
Afficher un message dans la barre d'état	754
Afficher des images	754
Les sons	755
Optimiser le chargement des applets à l'aide des fichiers d'archives	756
Les applets et la sécurité	757
Mettre en place les autorisations	758
Comment Java utilise les fichiers d'autorisations	762
Utiliser le security manager avec les applications	763
Accéder à un URL à partir d'une application	763
Conclusion	767

Chapitre 21 :

Présentation des Java-Beans	769
-----------------------------	-----

Le concept de Beans	769
Le Beans Development Kit	771
Installation du BDK	771
Utilisation de la BeanBox	774

Ajouter un bean dans la fenêtre de composition	775
Sélection d'un bean dans la fenêtre de composition	776
Déplacement d'un bean	777
Redimensionner un bean	777
Modifier les propriétés d'un bean	777
Installer un handler d'événement	780
Relier des propriétés	784
Enregistrer les beans	786
Générer une applet	786
Création d'un bean	787
Codage d'un bean	789
Création d'un conteneur de test	790
Packager un bean	791
Préparation du manifeste	792
Création de l'archive	793
Utilisation du bean dans la BeanBox	793
Résumé	

Chapitre 22 :

Les servlets et les Java Server Pages

	795
Qu'est-ce qu'une servlet ?	796
Structure et cycle de fonctionnement d'une servlet HTTP	797
Ce dont vous avez besoin	798
Installation du JSWDK 1.0.1	799
Configuration du chemin d'accès aux classes	799
Modification du fichier startserver.bat	800
Création d'un raccourci	800
Le répertoire webpages	801
Démarrage du serveur	801
Accès au serveur	801
Arrêter le serveur	803
Configuration du serveur	804
Test d'une servlet	805
Création d'une servlet	807

Configuration de la variable d'environnement classpath	808
Codage de la servlet	809
Utilisation de la première servlet	809
Description de la première servlet	810
Une servlet pour exploiter un formulaire	815
Codage de la servlet	816
Le formulaire	818
Utilisation du formulaire	820
Problèmes de synchronisation	823
Comparaison avec Perl	824
Les servlets pour remplacer les SSI	825
Un compteur d'accès	827
Les Java Server Pages	830
Directives	831
Déclarations	831
Scriptlets	831
Expressions	832
Le code généré	832
Résumé	835

Chapitre 23 :

Java et XML

Principes fondamentaux de XML	837
Utiliser un transformateur XSLT : Apache Xalan	843
Installation d'Apache Xerces et Apache Xalan	844
Utilisation de Xalan	845
Une servlet générant un fichier texte	847
Une servlet produisant un document RTF	854
Une servlet produisant des documents PDF	856
Installation de FOP	863
Codage de la servlet	863
Une servlet affichant plusieurs formats	873
Résumé	879

Chapitre 24 :

JDBC et les bases de données	881
Installation de MySQL sous Windows	882
Configuration de MySQL sous Windows	883
Configuration sous Windows NT	884
Installation de MySQL en tant que service	887
Test de MySQL sous Windows	888
Installation de MySQL sous Linux	891
Configuration de MySQL sous Linux	891
Test de MySQL sous Linux	892
Utilisation de MySQL	893
Noms d'utilisateurs et mots de passe	893
Création d'une base de données	896
Création d'une table et saisie de données	897
Installation d'un pilote JDBC	900
Test du pilote JDBC	903
En cas d'erreur	904
Analyse du programme	905
Lire les données	908
Connexion à un serveur distant	911
Une servlet pour consulter une base de données	911
Écriture dans une base de données à l'aide d'une servlet	919
Résumé	927

Annexe A :

Les classes String et StringBuffer	929
L'opérateur +	929
Les constructeurs de la classe String	930
La classe StringBuffer	933

Annexe B :

Les mots réservés	935
-------------------	-----

Annexe C :

Configuration de UltraEdit	937
----------------------------	-----

UltraEdit n'est pas gratuit !	939
Installation et configuration de UltraEdit	939

Index	945
-------	-----

AVAIENT D'AVOIR ÉCARTÉ CEI PEU, DANS L'ASSOCI, MAIS CEST UNE ÉTER-
NITÉ À L'ÉCHELLE DU DÉVELOPPEMENT DE L'INFORMATIQUE. QU'EN EST-IL
AUJOURD'HUI DES OBJETS QUI ONT PRÉCÉDÉ À SON LANCEMENT ?

Pourquoi Java ?

Java devait être un langage multi-plate-forme qui permettait, selon le principe préconisé par Sun Microsystems, son concepteur, d'écrire une fois pour toutes des applications capables de fonctionner dans tous les environnements. L'objectif était de taille, puisqu'il impliquait la définition d'une *machine virtuelle Java (JVM)* sur laquelle les programmes écrits devaient fonctionner, ainsi que la réalisation de cette machine virtuelle dans tous les environnements concernés. Sun Microsystems se chargeait par ailleurs de la réalisation d'une machine virtuelle dans les environnements Solaris et Windows¹, laissant à d'autres le soin d'en faire autant pour les autres environnements (et en particulier Mac OS, le système d'exploitation des

1. "Write Once, Run Anywhere" - écrire une fois, utiliser partout.

2. Dans le jargon informatique la définition est souvent appelée "spécification". De même, elle est réalisée dans un exécutable puis se trouve conforme à la spécification en s'exécutant.